

---

# GENETICKÝ ALGORITMUS PRO HRU HAD

---

David Mašek  
masekda4@fit.cvut.cz

## 1 Úvod

Cílem práce je vyšlehtit řídicí funkci pro hru had pomocí genetického programování a předvést její funkčnost. Nejdůležitější částí práce bylo především správné navržení genotypu a výběr jednotlivých operátorů a konstant.

Základní algoritmus:

---

```
population = init(POPULATION_SIZE)
generation = 0
log(population)
while generation < MAX_GENERATIONS do
    new_population = init()
    new_population = population.get_best()
    while len(new_population) < POPULATION_SIZE do
        (first, second) = population.select_two()
        if random() < CROSSOVER_RATE then
            first.crossover(second)
        end if
        if random() < MUTATION_CHANGE then
            first.mutate()
        end if
        first.prune()
        first.calculate_fitness()
        new_population.append(first)
        if len(new_population) < POPULATION_SIZE then
            ... repeat code above for second ...
        end if
    end while
    population = new_population
    log(population)
end while
```

---

## 2 Konstanty

V tabulce jsou vypsané použité konstanty, včetně stručných vysvětlivek. Pokud bude potřeba podrobnější vysvětlení, bude uvedeno dále v textu. Dále jsou u nich uvedeny experimentálně nalezené hodnoty. Případné odklony od zde uvedených hodnot jsou zdůrazněny.

Použité konstanty		
Jméno	Hodnota	Vysvětlivka
MAX_TURNS	5000	Počet kol, po kterém je ukončena simulace
MAX_TURNS_LOW	500	Počet kol, po kterém je ukončena simulace, pokud má had nízké skóre
LOW_SCORE	10	Hranice nízkého skóre
MAX_TURNS_ZERO	100	Počet kol, po kterém je ukončena simulace, pokud má had nulové skóre
RESTRICT_DEPTH	6	Omezuje délku stromu
BASE_MUTATION_RATE	0.05	Ovlivňuje míru mutace
MUTATION_CHANCE	0.05	Pravděpodobnost, že dojde k mutaci
CROSSOVER_RATE	0.05	Pravděpodobnost, že dojde ke křížení
GENERATIONS	80	Počet generací
POPULATION_SIZE	200	Velikost populace
TOURNAMENT_SIZE	0.05	Poměr jedinců (vzhledem k velikosti populace), kteří se účastní turnaje
WIDTH	30	Šířka herního pole, na kterém probíhá simulace
HEIGHT	20	Výška herního pole, na kterém probíhá simulace
NEARBY_DISTANCE	3	Vzdálenost vyhodnocená jako "blízko"

### 3 Genotyp

#### 3.1 Terminály

Jako terminály jsem zvolil rotace hada vzhledem k současnému směru (poslednímu pohybu). Terminály jsem si označil jako NONE, LEFT a RIGHT (tedy žádná rotace, doleva a doprava).

#### 3.2 Funkce

##### 3.2.1 Zvolené funkce

Jako funkce jsem zvolil "if-funkce" založené na inputu z herního světa. Funkce se skládá vždy ze tří částí:

- entita: had, jídlo, zeď, blok (had nebo zeď)
- míra vzdálenosti: AHEAD (na následujícím políčku), VISIBLE (na následujícím neobsazeném políčku) a později přidaná NEARBY (vzdálenost  $\leq$  NEARBY\_DISTANCE)
- typ rotace: LEFT, RIGHT, NONE (určuje, kterým směrem se dívat, vzhledem k současnému směru)

##### 3.2.2 Výběr funkcí

Zpočátku jsem měl funkce založené pouze na entitách blok a jídlo, ale evoluce neprodukovala jedince s dostatečně komplexním chováním. Přidání funkcí založených na detekci přímo zdi nebo hada vedlo k podstatnému zlepšení. Dalšího zlepšení, jsem chtěl dosáhnout přidáním další "úrovně" vzdálenosti - NEARBY. To přineslo jistou míru zlepšení, ale také podstatné zvýšení výpočetní náročnosti algoritmu.

##### 3.2.3 Další experimenty

Otestoval jsem i další možnosti funkcí. Příliš malé množství funkcí vedlo k příliš jednoduchému chování. Pokusil jsem se i výrazně zvětšit počet funkcí přidáním více "hladin" vzdálenosti, ale příliš velké množství funkcí podstatně snižovalo procento užitečných jedinců a evoluce měla problém vyprodukovat vůbec nějaké schopné jedince.

### 4 Operátory

#### 4.1 Inicializace

Použil jsem omezení hloubky stromu částečně založené na náhodě, kde pravděpodobnost, že vytvořený uzel bude terminál je  $depth/RESTRICT\_DEPTH$  (depth je současná hloubka zanoření).

## 4.2 Mutace, křížení a prořezávání

Konstanta `MUTATION_CHANCE` určuje šanci s jakou se zavolá metoda mutace. Ta pak prochází strom a na každém uzlu provede mutaci s pravděpodobností  $BASE\_MUTATION\_RATE * (depth + 1)$ . Pokud dojde v nějakém uzlu k mutaci, pokračuje v mutaci v potomcích s pravděpodobností `MUTATION_CHANCE`. Zavedením dvou konstant pro mutaci, jsem chtěl dosáhnout toho, že budou k mutaci pravděpodobněji zvoleny hlouběji umístěné uzly.

Provedená mutace je vždy nahrazení uzlu náhodným stromem (viz operátor inicializace).

Jako metodu křížení jsem zvolil prohození dvou náhodných podstromů.

Dále jsem zavedl základní prořezávání, kde jsou funkce, které vrací v obou případech stejné terminály nahrazeny přímo oním terminálem, což zjednodušuje výsledné stromy.

## 4.3 Fitness

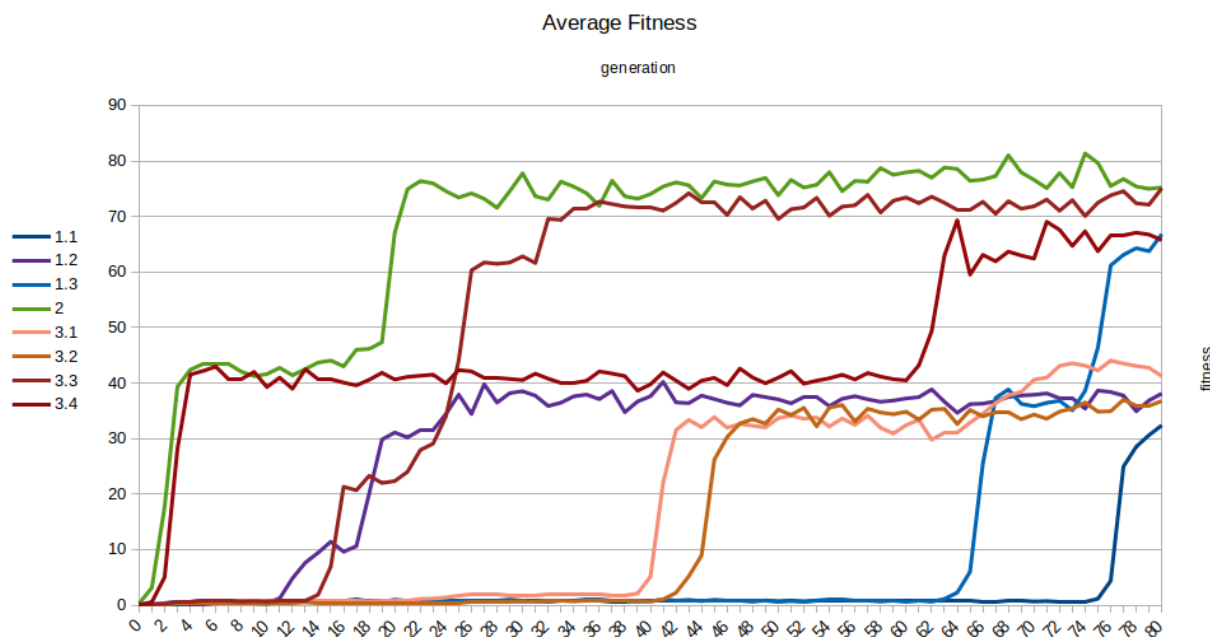
Fitness je počítána pomocí simulace hry s daným rozhodovacím stromem. V každé hře je počítáno skóre a počet tahů. Z důvodů proveditelnosti bylo nutné zavést maximální počet tahů (viz konstanty). Rozhodl jsem se zavést těchto konstant několik (v závislosti na skóre), aby neužitečné hry, kdy had např. běhá dokola byly ukončeny rychleji a zároveň se úspěšnější jedinci měli možnost předvést. Prvotní způsob počítání fitness -  $fitness = score + \frac{turns}{MAX\_TURNS}$  jsem nahradil  $fitness = score + \frac{score}{turns}$  za účelem preference aktivnějších jedinců, což se osvědčilo.

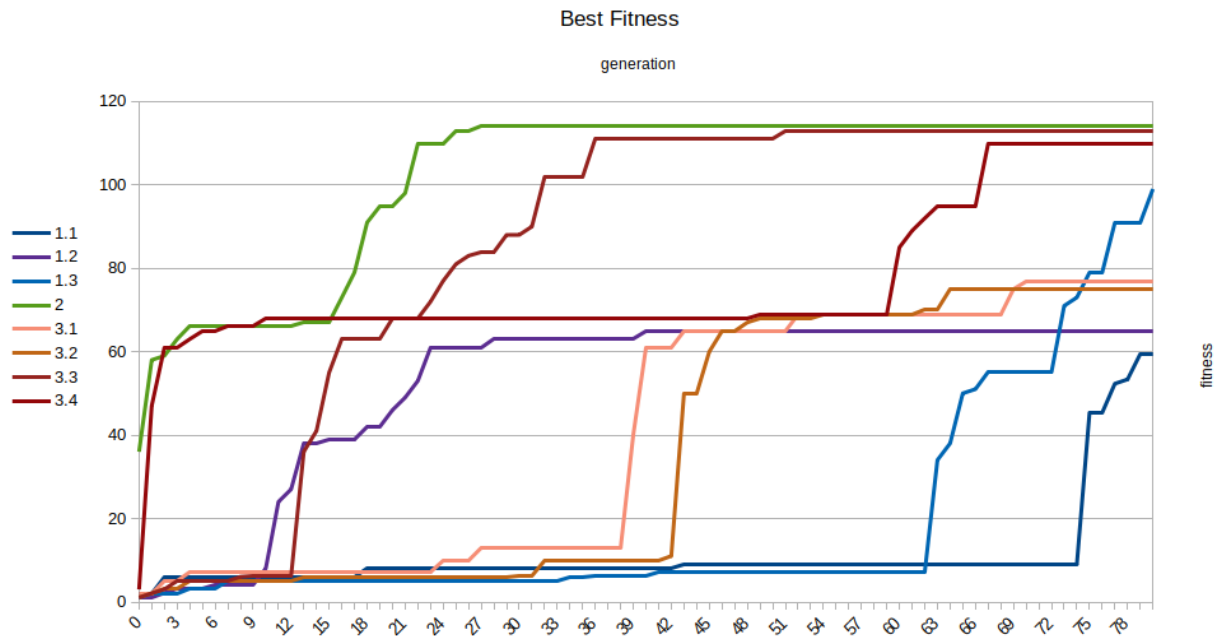
# 5 Výsledky

## 5.1 Ukázka průběhu - fitness

Následuje ukázka vývoje fitness. Ukázány jsou tři mírně odlišné konfigurace (formát je "číslo\_konfigurace.běh"). První dvě obsahují pouze funkce založené na "VISIBLE" a "AHEAD". Třetí přidává funkce "NEARBY". Jejich odlišnosti od hodnot uvedených v části konstanty:

- 1. `WIDTH = 25, DEPTH = 18, CROSSOVER_RATE = 0.1, MAX_TURNS = 4000, RESTRICT_DEPTH = 5, POPULATION = 100`
- 2. `RESTRICT_DEPTH = 4`
- 3. viz sekce konstanty





## 5.2 Závěr

Povedlo se vyšlechtit hada, který hraje hru poměrně dobře, vyhýbá se překážkám a relativně efektivně sbírá jídlo. Při větší délce se však dostává do situací, které nutně vedou k jeho záhubě. Je tu stále velký prostor pro zlepšení.

Průměrný jedinec se pozvolna zlepšuje, dokud se evoluci nepodaří vyšlechtit několik "dobrých" jedinců. Průměrná fitness pak rychle narůstá, dokud nedorazí do bodu, kde už stagnuje a pouze výjimečně se jí podaří vygenerovat lepší jedince.

## 5.3 Reference

- materiály předmětu BI-ZUM na FIT ČVUT <https://courses.fit.cvut.cz/BI-ZUM/>